

Listing of claims:

The following listing of claims replaces all previous claim listings in the application:

1. **(Currently Amended)**. A method for transferring a set of files, the method comprising:
  - receiving, at a destination fileserver, ~~metadata and~~ a set of stub files associated with the set of files;
  - ~~maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver; that are associated with each file in the set of files by updating a location component in a fileserver, wherein said repository nodes store a replica of said file; and,~~
  - using the lists, initiating recovery of files in the set of files on the destination fileserver;
  - using a stub file in the set of stub files, allowing access to a full content of a file associated with the stub file;
  - replacing each stub file with a full content of the file associated with the stub file;
  - and
  - wherein said replacing includes
    - receiving a client request for a specified file in the set of files;
    - replacing the stub file associated with the specified file with a full content of the specified file.

2. (Previously Presented). The method of claim 1 wherein the metadata is received at said destination fileserver from a repository node.

3. (Previously Presented). The method of claim 1 further comprising:  
selecting said destination fileserver for receiving said metadata and said stub files.

4. (Previously Presented). The method of claim 1 further comprising:  
selecting a share of data for receiving at said destination fileserver.

5. (Original). The method of claim 1 wherein the set of files is the set of files that have been accessed during a specified period and wherein replacing each stub file comprises recursively replacing the stub file associated with the file that was most-recently accessed until all the stub files in the set of files have been replaced.

6. (Original). The method of claim 5 wherein the specified period is a most-recent period.

7. (Original). The method of claim 1 wherein the location component is a location cache.

8. (**Currently Amended**). A data protection system comprising:

a fileserver having:

a file system operative to store client files;

a fileserver API operative to communicate with a repository;

a fileserver file transfer module in communication with the file system and ~~operative~~ configured to transfer files for the file system to and/or from at least one repository;  
and

a recovery service in communication with the fileserver API and with the file system and ~~operative~~ configured to transfer a set of files, the recovery service having:

a receiving component ~~operative~~ configured to receive metadata and stub files associated with the set of files at the fileserver;

a location updating component in communication with the receiving component and ~~operative~~ configured to maintain a list of repository nodes that contain a replica of each file in the set of files and a list of files in the set of files stored at the destination fileserver that are associated with each file in the set of files, wherein said repository nodes store a replica of said file; and

wherein using the lists, said recovery service is configured to initiate recovery of files in the set of files on the fileserver;

wherein using a stub file in the set of stub files, said recovery service is configured to allow access to a full content of a file associated with the stub file; and

a stub file replacement component in communication with the receiving component and ~~operative~~ configured to replace each stub file with the full content of the file associated with the stub file.

9. (Currently Amended). The system of claim 8 further comprising

a filter driver operative to intercept input/output activity initiated by client file requests and to maintain a list of modified and created files since a prior backup;

a policy cache operative to store a protection policy associated with a share;

a mirror service in communication with the filter driver and with the policy cache, the mirror service ~~operative~~ configured to prepare modified and created files in a share to be written to a repository as specified in the protection policy associated with the share.

10. **(Currently Amended)**. The system of claim 9 further comprising:

a location cache in communication with the mirror service and ~~operative~~ configured to indicate which repository should receive an updated version of an existing file; and

a location manager coupled to the location cache and ~~operative~~ configured to update the location cache when the system writes a new file to a specific repository node.

11. **(Currently Amended)**. The system of claim 8 further comprising

a local repository having:

a local repository node API ~~adapted for~~ configured to communicating with the fileserver API;

a local repository file transfer module in communication with the fileserver file transfer module and ~~adapted for~~ configured to transferring files to the fileserver file transfer module; and

a data mover in communication with the local repository API and ~~operative~~ configured to supervise the replication of files from the local repository to the fileserver.

12. **(Currently Amended)**. The system of claim 11 wherein the fileserver API is ~~operative~~ configured to communicate with a network and wherein the system further comprises:

a remote repository having:

a remote repository node API ~~adapted for~~ configured to communicating with the network;

a remote repository file transfer module in communication with the local file transfer module and ~~adapted for~~ configured to transferring files to the fileserver file transfer module; and

a data mover in communication with the remote repository API and ~~operative~~ configured to supervise the replication of files from the remote repository to the fileserver.

13. **(Currently Amended)**. A method for storing data, the method comprising:

providing a fileserver having:

a file system operative to store client files;

a policy component ~~operative~~ configured to store a protection policy associated with a set of files;

a mirror service in communication with the policy component, the mirror service operative to prepare modified and created files in a set of files to be written to a repository as specified in the protection policy associated with the set of files;

a fileserver API coupled to the mirror service and ~~operative~~ configured to communicate with a repository;

a fileserver file transfer module in communication with the file system and ~~operative~~ configured to transfer files for the file system to and/or from at least one repository;  
and,

a location updating component ~~operative~~ configured to maintain a list of repository nodes that contain a replica of each file in the set of files and a list of files in the set of files stored at the destination fileserver ~~that are associated with each file in the set of files;~~  
~~wherein said repository nodes store a replica of said file;~~

wherein using the lists, said fileserver is configured to initiate recovery of files in the set of files on the fileserver;

wherein using a stub file in the set of stub files, said fileserver is configured to allow access to a full content of a file associated with the stub file;

determining a caching level for said fileserver ~~as stored in the policy component;~~  
and

recursively, determining a utilization of the fileserver;  
comparing the caching level against the utilization; and  
creating a file migration candidate list when the utilization exceeds the caching level;

staging out one candidate file;  
replacing the candidate file with a stub file; and  
determining whether the utilization of the fileserver still exceeds the caching level.

14. (Previously Presented). The method of claim 13 wherein said determining if the utilization of the fileserver still exceeds the caching level further comprises staging out another candidate file on the candidate list and again determining if the utilization of the fileserver exceeds the caching level.

15. (Previously Submitted). The method of claim 1, wherein said replacing the stub file for the specified file is a higher priority task than replacing the stub files for non-requested files.